<u>In the Claims:</u>

This listing of claims will replace all prior version and listings of claims in the application:

1 -18 Cancelled.

19. (previously presented) A peer-vector machine, comprising:
a buffer;
a bus;
a processor coupled to the buffer and to the bus and operable to;
        execute an application, first and second data-transfer objects, and a communication object,
        publish data under the control of the application,
        load the published data into the buffer under the control of the first data-transfer object,
        retrieve the published data from the buffer under the control of the second data-transfer object,
        construct a message under the control of the second data-transfer object, the message including the retrieved published data and information indicating a destination of the retrieved published data, and
        drive the message onto the bus under the control of the communication object; and
a pipeline accelerator coupled to the bus, including the destination, and operable to receive the message from the bus, to recover the received published data from the message, to provide the recovered data to the destination, and to process the recovered data at the destination without executing a program instruction.

20. (previously presented) The peer-vector machine of claim 19 wherein the destination includes a field-programmable gate array that is operable to process the recovered data.

21. (previously presented) The peer-vector machine of claim 19, further comprising:

a registry coupled to the processor and operable to store object data; and

wherein the processor is operable to;

execute an object factory, and

generate the first and second data-transfer objects and the communication object from the object data under the control of the object factory.

22. (previously presented) A peer-vector machine, comprising:

a buffer;

a bus;

a pipeline accelerator coupled to the bus and operable to generate data without executing a program instruction, to generate a header including information indicating a destination of the data, to package the data and header into a message, and to drive the message onto the bus; and

a processor coupled to the buffer and to the bus and operable to:

execute an application, first and second data-transfer objects, and a communication object,

receive the message from the bus under the control of the communication object,

load into the buffer, under the control of the first data-transfer object, the received data without the header, the buffer corresponding to the destination of the data,

unload the data from the buffer under the control of the second data-transfer object, and

process the unloaded data under the control of the application.

23. (previously presented) The peer-vector machine of claim 22 wherein the processor is operable to:

receive the message from the bus under the control of the communication object; and

recover the data from the message under the control of the first data-transfer object.

24. (previously presented) The peer-vector machine of claim 22, further comprising:

a registry coupled to the processor and operable to store object data; and

wherein the processor is operable to,

execute an object factory, and

to generate the first and second data-transfer objects and the communication object from the object data under the control of the object factory.

25 – 50 Cancelled.

51. (previously presented) A method, comprising:

publishing data with an application running on a processor;

loading the published data into a buffer with a first data-transfer object running on the processor;

retrieving the published data from the buffer with a second data-transfer object running on the processor;

generating information that indicates a hardwired pipeline for processing the retrieved data;

packaging the retrieved data and the information into a message;

driving the message onto a bus with a communication object running on the processor;

receiving the message from the bus; and

processing the published data with the indicated hardwired pipeline without executing a program instruction, the indicated hardwired pipeline being part of a pipeline accelerator that includes a field-programmable gate array.

52. (previously presented) The method of claim 51 wherein:

packaging the retrieved data and the information into a message comprises generating the message including a header, and the published data, with the second data-transfer object;

driving the data onto the bus comprises driving the message onto the bus with the communication object; and

receiving the published data comprises receiving the message and recovering the published data from the message with the pipeline accelerator.

53. (previously presented) A method, comprising:

generating, with a pipeline accelerator and without executing a program instruction, a message header that includes a destination of data, the destination identifying a software application for processing the data;

generating, with the pipeline accelerator and without executing a program instruction, a message that includes the header and the data;

driving the message onto a bus with the pipeline accelerator;

receiving the message from the bus with a communication object running on a processor;

loading into a buffer, with a first data-transfer object running on the processor, the received data absent the header, the buffer being identified by the destination;

unloading the data from the buffer with a second data-transfer object running on the processor; and

processing the unloaded data with the software application running on the processor.

54. (previously presented) The method of claim 53, further comprising recovering the data from the message with the first data-transfer object.

55 – 61 Cancelled.

62. (previously presented) A peer-vector machine, comprising:
a buffer;

a single bus;

a processor coupled to the buffer and to the bus and operable to:

execute an application, first and second data-transfer objects, and a communication object,

publish data under the control of the application,

load the published data into the buffer under the control of the first data-transfer object,

retrieve the published data from the buffer under the control of the second data-transfer object,

construct a message under the control of the second data-transfer object, the message including the retrieved published data and information indicating a destination of the retrieved published data, and

drive the message onto the bus under the control of the communication object; and

a pipeline accelerator coupled to the bus, including the destination, and operable to receive the message from the bus, to recover the received published data from the message, to provide the recovered data to the destination, and to process the recovered data at the destination without executing a program instruction.